This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1.   (Currently Amended) A computer-implemented method of communicating with an application using an application programming interface (API), the method executing in a computer system having M front-end languages and N back-end compilers, where M and N are integers, the API method comprising:

the system receiving from the application, one or more calls to set one or more compile parameters and commands for converting a plurality of input queries in a plurality of language compilers to an XML intermediate language representation, wherein the XML intermediate language representation is a composite of the plurality of input queries, is an explicit representation of the meaning of the plurality of input queries, and has a graph structure; and

the system receiving from the application, one or more calls to convert the XML intermediate language representation to at least one query, the at least one query, when executed, instructing the system to query over a plurality of data sources having differing data models;

wherein the one or more calls to convert the XML intermediate language to at least one query comprises a plurality of API calls comprising commands to a plurality of target generators and corresponding execution engines having different data languages, wherein the plurality of API calls share a common data model comprising operators of the graph structure of the XML intermediate data language representation, and wherein use of the M front-end languages and the N back-end compilers results in only M plus N system paths instead of M times N system paths.

2.   (Original) The method of claim 1, further comprising the application receiving from the system one or more of the group consisting of event status, progress status, intermediate results, final results, error messages, warnings and help messages.

3.   (Original) The method of claim 1, wherein the one or more calls to set one or more environment, compile parameters and compile commands comprise one or more of enabling message reception from the system, specifying query permission and execution restrictions, selecting the input query and compiler type, and establishing evaluation contexts.

4.   (Original) The method of claim 3, wherein the compiler type comprises XPath, XSLT and XQuery language compilers.

5.   (Original) The method of claim 1, wherein the XML intermediate language representation is a semantic representation of an input query.

6.   (Previously Presented) The method of claim 1, wherein converting the XML intermediate language to the executable query comprises preparing the XML intermediate language for direct execution in a target query execution engine, wherein direct execution avoids the use of a compiler for the target execution engine.

7.   (Original) The method of claim 1, wherein converting the XML intermediate language to the executable query comprises converting the XML intermediate language into a target representation using a target generator.

8.   (Original) The method of claim 7, wherein the target representation is one or more of the group consisting of an XML language target, a SQL language target and an intermediate language target.

9.   (Cancelled)

10.   (Currently Amended) A system for compilation and execution of input queries producing query results, comprising:

        a plurality of input devices for receiving a plurality of input queries;

~~one or more~~ a quantity of M front-end query language to intermediate language compilers wherein a composite XML intermediate language representation is compiled from the plurality of input queries, wherein the XML intermediate language representation is an explicit representation of the meaning of the plurality of input queries and has a graph structure;

~~a plurality of~~ a quantity of N back-end target generators wherein the XML intermediate language representation is transformed into a plurality of target queries;

a plurality of data sources for querying over; and

a plurality of execution engines wherein the plurality of target queries are executed over the plurality of data sources to produce the query results; and

wherein use of the M front-end query language to intermediate language compilers and the N back-end target generators results in only M plus N system paths instead of M times N system paths.

11.   (Previously Presented) The system of claim 10, wherein the plurality of input queries comprise queries formed from one or more of XPath, XSLT, and XQuery languages.

12.   (Original) The system of claim 10, wherein the XML intermediate language representation expresses the meaning of the input query.

13.   (Previously Presented) The system of claim 10, wherein the plurality of target generators comprise one or more of an XML language generator, a SQL language generator and an intermediate language generator.

14.   (Previously Presented) The system of claim 10, wherein the plurality of data sources comprise one or more of relational data sources and non-relational data sources.

15.   (Original) The system of claim 14, wherein non-relational data sources comprise spreadsheets and word processing documents.